UTS
UNIVERSITY OF TECHNOLOGY SYDNEY

The Architecture of a Quantum Programming Environment

by

Shusen Liu

A thesis submitted in partial fulfilment of the
requirements for the degree of Doctor of Philosophy

Supervisor: Prof. Runyao Duan

Co-supervisor: Prof. Mingsheng Ying

at the
Centre for Quantum Software and Information
Faculty of Engineering and Information Technology
University of Technology Sydney

October 2018

Certificate of Original Authorship

I, *Shusen Liu* declare that this thesis, is submitted in fulfilment of the requirements for the award of *Doctor of Philosophy Degree*, in *Software Engineering at the Faculty of Engineering and Information Technology* at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise reference or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This thesis is the result of a research candidature conducted jointly with Sun Yat-sen University as part of a collaborative doctoral degree.

This document has not been submitted for qualifications at any other academic institution.

Signed:

Production Note:
Signature removed prior to publication.

Date:

October 2018

The Architecture of a Quantum Programming Environment

by

Shusen Liu

A thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy

Abstract

This thesis presents the architecture of quantum programming environment, called QSI, along with its related modules and several quantum experiments. The environment is based on one specific quantum language, namely quantum **while**-language. Some partial experimental results are also presented within QSI.

The first part relates to the architecture, the designing and the implementation of quantum programming environment which provides a new, powerful and flexible environment for developing and implementing quantum programs. First, we study the possible structure of the programming environment which supports a measurement-based case statement and a measurement-based **while**-loop. These two program constructs are extremely convenient for describing large-scale quantum algorithms, such as quantum random walk-based algorithms. We also define a new assembly language called f-QASM (Quantum Assembly Language with feedback) as an interactive command set. The assembly language is compatible with other low-level instruction sets and can be used to directly drive quantum hardware. Moreover, the simulation of syntax of quantum program and the behaviours within the architecture on the classical computer are discussed. Finally, we consider the work-flow which contains the decomposition of unitary matrix to achieve the goal that executing on Noisy Intermediate-Scale Quantum Computer.

The second part concerns the modules based on quantum programming environment: termination analysis module, detective separable unitary module and quantum control module. Along with the architecture, we bring an essential module - termination analysis

module for the loop structure. It can analyze sub-bodies of quantum program and suggest the critical termination information. In addition, we improve the Jordan decomposition step in the original algorithm which consumes extended period for analyzing. This improvement also makes the module more robust on executing. A fast permutation algorithm module clarifies the re-ordering algorithm in case of qubits system. It regenerates the program (unitary operator) which is not in pre-ordered sequence. In the detective separable unitary module, we prove sufficient conditions for separable unitary and its approximate scenario. The result shows there does not exist a universal algorithm for potential parallel executing quantum programs without communications (classical or quantum communications). However, in approximate, there exists a scheme for parallel computing without the help of communication. In this part, two examples for parallel computing are given. Last, in quantum control module, an algorithm is suggested towards automatically generating quantum circuits for quantum case-statement. We believe these analysis modules can help the compiler to optimize the implementation of quantum algorithms.

The third part is devoted to quantum experiment. First, we focus several experiments which can be operated directly by QSI : Qloop, BB84 protocol and Grover search algorithm. After that, with the help of IBM's QISKit, two impressive experiments: distinguishing unitary gates and Bell states are given on real quantum computer. Finally, we combine QSI with Microsoft's LIQU$i|\rangle$ to implement quantum case-statement. These experiments significantly show the quantum power and the scalable framework of the quantum programming environment in practice.

Acknowledgements

The research of this dissertation could not have been performed if not for the assistance, patience, and support of many individuals. First, I would like to express my sincere gratitude to my supervisor Prof. Runyao Duan for supervising my research. His insightful and rigorous instruction has motivated me to constantly purse studies in the quantum area. His encouragement to not give up when the enormous scope of QSI become too overwhelming was my last resource for achieving this goal. It sincerely touched me that he even contributed an amount some of his own time to debugging and the details of the designing. Also, I wish to thank Prof. Mingsheng Ying for introducing me to the topic and for his support on the way. His knowledge and understanding broadened my views on quantum programming and guided me in the right direction.

I would additionally extend my deepest gratitude to Mr Yang He for sharing his wealth of experience in classical programming with me and saving me with some programming techniques. Without the wisdom and skills he has accumulated over the last two decades, our programming environment would not have been developed so quickly.

I also want to show my thanks to Prof. Kan He. The routine discussions during his visit to Sydney illuminated my thinking on fundamental knowledge and made the 'impossible' task of designing the compiler possible.

I also express my thanks to Jemima Moore, who spends much time on proofreading this thesis after final examination. She examines the entire thesis carefully and thoroughly and gives many useful suggestions.

Further, I wish to thank my parents for supporting me as a higher degree research student. They provided me with all their love and care and asked for nothing in return.

Finally, and in particular, I wish to extend my thanks to my love, Ms Huanhua Zhang. She has spent two years with me and fully devoted herself to supporting my research.

Contents

List of Figures

List of Tables

List of Publications

[1] Shusen Liu; Yinan Li; Runyao Duan. Distinguishing unitary gates on the ibm quantum processor (accepted as regular paper). *SCIENCE CHINA Information Sciences*, 2018.

[43] Shusen Liu, Xin Wang, Li Zhou, Ji Guan, Yinan Li, Yang He, Runyao Duan, and Mingsheng Ying. Qsi: a quantum programming environment. *arXiv preprint arXiv:1710.09500*, 2017.

[3] Shusen Liu, Yang He, and Cai Zhang. Towards automatically construct quantum circuits for quantum programs with quantum control. In *2017 IEEE 85th Vehicular Technology Conference (VTC Spring)*, pages 1–5, June 2017. doi: 10.1109/VTC-Spring.2017.8108699.